

# Domain decomposition algorithms for the compressible Euler equations

V. Dolean\*, F. Nataf†

15th September 2005

## Abstract

In this work we present an overview of some classical and new domain decomposition methods for the resolution of the Euler equations. The classical Schwarz methods are formulated and analyzed in the framework of first order hyperbolic systems and the differences with respect to the scalar problems are presented. This kind of algorithms behave quite well for bigger Mach numbers but we can further improve their performances in the case of lower Mach numbers. There are two possible ways to achieve this goal. The first one implies the use of the optimized interface conditions depending on a few parameters that generalize the classical ones. The second is inspired from the Robin-Robin preconditioner for the convection-diffusion equation by using the equivalence via the Smith factorization with a third order scalar equation.

## 1 Introduction

When solving the compressible Euler equations by an implicit scheme the nonlinear system is usually solved by Newton's method. At each step of this method we have to solve a linear system which is non-symmetric and very ill conditioned. The necessity of a domain decomposition became more and more obvious. In a previous paper [DLN04] we formulated a Schwarz algorithm (interface iteration which relies on the successive solving of the local decomposed problems and the transmission of the result at the interface) involving transmission conditions that are derived naturally from a weak formulation of the underlying boundary value problem (first formulated in [QS96]). As far as these algorithms are concerned, when dealing with supersonic flows, whatever the space dimension is, imposing the appropriate characteristic variables as interface conditions leads to a convergence of the algorithm which is optimal with regards to the number of subdomains. The only case of interest remains the subsonic one where this property is lost except in the one dimensional case. We recall briefly these results in order to introduce more general and performant methods such as the optimized interface conditions and the preconditioning methods. The former were widely studied and analyzed for scalar problems such as elliptic equations in [Lio90, EZ98], for the Helmholtz equation in [BD97, CN98] convection-diffusion problems in [JNR01]. For time dependent problems and local times steps, see for instance [GHN01]. The preconditioning methods have also known a wide developpement in the last decade. The Neumann-Neumann algorithms for symmetric second order problems [RT91] has been the subject of numerous works, see [TW04] and references therein. An extension of these algorithms to non-symmetric scalar problems (the so called Robin-Robin algorithms) has been done in [ATNV00, GGTN04] for advection-diffusion problems.

In the section 2 we first formulate the Schwarz algorithm for a general linear hyperbolic system of PDEs with general interface conditions inspired by Clerc [DN04b] built in order to have a well-posed problem. The convergence rate is computed in the Fourier space as a function of some parameters. We will further

---

\*Univ. de Nice and INRIA, Sophia Antipolis, 06108 Nice Cedex 02, France, dolean@math.unice.fr

†CMAP, CNRS UMR 7641, Ecole Polytechnique, 91128 Palaiseau Cedex, France, nataf@cmap.polytechnique.fr

estimate the convergence rate at the discrete level. We will find the optimal parameters of the interface conditions at the discrete level. We will then use the new optimal interface conditions in Euler computations which illustrate the improvement over the classical interface conditions (first described in [QS96]).

As far as preconditioning methods are concerned, to our knowledge, no extension to the Euler equations was done. In the section 2 we will first show the equivalence between the 2D Euler equations and a third order scalar problem, which is quite natural by considering a Smith factorization of this system, see [Gan66] then we define an optimal algorithm for the third order scalar equation inspired from the idea of the Robin-Robin algorithm [ATNV00] applied to a convection-diffusion problem. Afterwards we back-transform it and define the corresponding algorithm applied to the Euler system. All the previous results have been obtained at the continuous level and for a decomposition into 2 unbounded subdomains. After a discretization in a bounded domain we cannot expect that these properties to be preserved exactly. Still we can show by a discrete convergence analysis that the expected results should be very good.

## 2 A Schwarz algorithm with general interface conditions

In this section introduce a Schwarz algorithm which is based on general transmission conditions at subdomain interfaces that take into account the hyperbolic nature of the problem. In addition, we recall some existing results concerning the convergence of the algorithm. We consider here a general non-linear system of conservation laws. Under the hypothesis that the solution is regular, we can also write it under a non-conservative (or quasi-linear) equivalent form:

$$(1) \quad \frac{\partial W}{\partial t} + \sum_{i=1}^d A_i(W) \frac{\partial W}{\partial x_i} = 0$$

where the  $A_i$  are the Jacobian matrices of the flux vectors. Assume that we first proceed to an integration in time of (1) using a backward Euler implicit scheme involving a linearization of the flux functions and eventually we symmetrize it (we know that when the system admits an entropy it can be symmetrized by multiplying it by the hessian matrix of this entropy). This operation results in the linearized system:

$$(2) \quad \mathcal{L}(\delta W) \equiv \frac{\text{Id}}{\Delta t} \delta W + \sum_{i=1}^d A_i \frac{\partial \delta W}{\partial x_i} = f$$

In the following we will define the boundary conditions that have to be imposed when solving the problem on a domain  $\Omega \subset \mathbb{R}^d$ . We denote by  $A_{\mathbf{n}} = \sum_{i=1}^d A_i n_i$ , the linear combination of jacobian matrices by the components of the outward normal vector at the boundary of the domain  $\partial\Omega$ . This matrix is real, symmetric and can be diagonalized

$$A_{\mathbf{n}} = T \Lambda_{\mathbf{n}} T^{-1}, \Lambda_{\mathbf{n}} = \text{diag}(\lambda_i)$$

It can also be splitted in negative and positive part using this diagonalization

$$A_{\mathbf{n}} = A_{\mathbf{n}}^+ + A_{\mathbf{n}}^-, A_{\mathbf{n}}^{\pm} = T \Lambda_{\mathbf{n}}^{\pm} T^{-1}, \Lambda_{\mathbf{n}}^+ = \text{diag}(\max(\lambda_i, 0)), \Lambda_{\mathbf{n}}^- = \text{diag}(\min(\lambda_i, 0))$$

This corresponds to a decomposition with local characteristic variables. A more general splitting in negative(positive) definite parts,  $A_{\mathbf{n}}^{neg}$  and  $A_{\mathbf{n}}^{pos}$  of  $A_{\mathbf{n}}$  can be done such that these matrices satisfy the following properties:

$$(3) \quad A_{\mathbf{n}} = A_{\mathbf{n}}^{neg} + A_{\mathbf{n}}^{pos}, \text{rank}(A_{\mathbf{n}}^{neg, pos}) = \text{rank}(A_{\mathbf{n}}^{\pm}), A_{\mathbf{n}}^{pos} = -A_{\mathbf{n}}^{neg}$$

In the scalar case the only possible choice is  $A_{\mathbf{n}}^{neg} = A_{\mathbf{n}}^-$ . Using the previous formalism we can define the following boundary condition:

$$A_{\mathbf{n}}^{neg} W = A_{\mathbf{n}}^{neg} g, \text{ on } \partial\Omega$$

**Remark 1** In the case of a classical decomposition in negative and positive part this boundary condition has the physical meaning of the incoming flux in domain  $\Omega$ . By extension of the properties found in this case we call the last equality of (3) conservation property because it insures that the “out-flow” quantity (given by the positive part of the jacobian flux matrix with opposite direction of the normal) is retrieved out of the “in-flow” quantity imposed by the boundary condition (given the negative part of the jacobian flux matrix).

## 2.1 Schwarz algorithm with general interface conditions

We consider a decomposition of the domain  $\Omega$  into  $N$  overlapping or non-overlapping subdomains  $\bar{\Omega} = \bigcup_{i=1}^N \bar{\Omega}_i$ . We denote by  $\mathbf{n}_{ij}$  the outward normal to the interface  $\Gamma_{ij}$  between  $\Omega_i$  and a neighboring subdomain  $\Omega_j$ . Let  $W_i^{(0)}$  denote the initial approximation of the solution in subdomain  $\Omega_i$ . A general formulation of a Schwarz algorithm for computing  $(W_i^{p+1})_{1 \leq i \leq N}$  from  $(W_i^p)_{1 \leq i \leq N}$  (where  $p$  defines the iteration of the Schwarz algorithm) reads :

$$(4) \quad \begin{cases} \mathcal{L}W_i^{p+1} &= f & \text{in } \Omega_i \\ A_{\mathbf{n}_{ij}}^{neg} W_i^{p+1} &= A_{\mathbf{n}_{ij}}^{neg} W_j^p & \text{on } \Gamma_{ij} = \partial\Omega_i \cap \Omega_j \\ A_{\mathbf{n}_{ij}}^{neg} W_i^{p+1} &= A_{\mathbf{n}_{ij}}^{neg} g & \text{on } \partial\Omega \cap \partial\Omega_i \end{cases}$$

where  $A_{\mathbf{n}_{ij}}^{neg}$  and  $A_{\mathbf{n}_{ij}}^{pos}$  satisfy (3). We have the following result concerning the convergence of the Schwarz algorithm in the non-overlapping case, due to ([Cle98]):

**Theorem 1** If we denote by  $E_i^p = W_i^p - W_i$  the error vector associated to the restriction to the  $i$ -th subdomain of the global solution of the problem. Then, the Schwarz algorithm converges in the following sense :

$$\begin{cases} \lim_{p \rightarrow \infty} \|E_i^p\|_{L^2(\Omega_i)^q} = 0 \\ \lim_{p \rightarrow \infty} \left\| \sum_{j=1}^d A_j \partial_j E_i^p \right\|_{L^2(\Omega_i)^q} = 0 \end{cases}$$

The convergence rate of the algorithm defined by (4) depends of the choice of the decomposition of  $A_{\mathbf{n}_{ij}}$  into a negative and a positive part satisfying (3). In order to choose the decomposition (3) we need to relate this choice to the convergence rate of (4).

## 2.2 Convergence rate of the algorithm with general interface conditions

We consider a two-subdomain non-overlapping or overlapping decomposition of the domain  $\Omega = \mathbb{R}^d$ ,  $\Omega_1 = ]-\infty, \gamma[ \times \mathbb{R}^{d-1}$  and  $\Omega_2 = ]\beta, \infty[ \times \mathbb{R}^{d-1}$  with  $\beta \leq \gamma$  and study the convergence of the Schwarz algorithm in the subsonic case. A Fourier analysis applied to the linearized equations allows us to derive the convergence rate of the “ $\xi$ ”-th Fourier component of the error. We will first briefly recall the technique of Fourier transform which was already described in detail in [DLN04]. The vector of Fourier variables is denoted by  $\boldsymbol{\xi} = (\xi_j, j = 2, \dots, d)$ . Let  $(E_i^p)(x) = (W_i^p - W_i)(x)$  be the error vector in the  $i$ th subdomain at the  $p$ th iteration of the Schwarz algorithm and:

$$\hat{E}(x_1, \xi_2, \dots, \xi_d) = \mathcal{F}E(x_1, \xi_2, \dots, \xi_d) = \int_{\mathbb{R}^{d-1}} e^{-i\xi_2 x_2 - \dots - i\xi_d x_d} E(x_1, \dots, x_d) dx_2 \dots dx_d$$

the Fourier symbol of the error vector. This transformation is useful only if the  $A_i$  matrices are constant which is the case here because we have considered the linearized form of the Euler equations around a constant state  $\bar{W}$ . The Schwarz algorithm in the Fourier space ( $\xi \in \mathbb{R}^{d-1}$ ) can be written as follows:

$$(5) \quad \begin{cases} \frac{d}{dx_1} \hat{E}_1^{p+1} &= -M(\boldsymbol{\xi}) \hat{E}_1^{p+1}, x < \gamma \\ \mathcal{A}^{neg}(\hat{E}_1^{p+1}) &= \mathcal{A}^{neg}(\hat{E}_2^p), \text{ on } x = \gamma \end{cases} \quad \begin{cases} \frac{d}{dx_1} \hat{E}_2^{p+1} &= -M(\boldsymbol{\xi}) \hat{E}_2^{p+1}, x > \beta \\ \mathcal{A}^{pos}(\hat{E}_2^{p+1}) &= \mathcal{A}^{pos}(\hat{E}_1^p), \text{ on } x = \beta \end{cases}$$

where we denoted by  $\mathcal{A}^{neg} = A_{\mathbf{n}}^{neg}$ ,  $\mathcal{A}^{pos} = A_{\mathbf{n}}^{pos}$  with  $\mathbf{n} = (1, 0)$  the outward normal to the domain  $\Omega_1$  and:

$$(6) \quad M(\boldsymbol{\xi}) = A_1^{-1} \left( \frac{1}{\Delta t} \text{Id} + \sum_{i=2}^d A_i \xi_{i-1} \right)$$

We thus obtain local problems that for a given  $\boldsymbol{\xi}$  are very simple ODEs whose solutions can be expressed as linear combinations of the eigenvectors of  $M(\boldsymbol{\xi})$  (we denote by  $\lambda_j(\boldsymbol{\xi})$  the eigenvalues of  $M(\boldsymbol{\xi})$ ). Here we require that these solutions are bounded at infinity ( $-\infty$  and  $+\infty$  respectively). We deduce that in the decomposition of  $\hat{E}_1(x_1, \boldsymbol{\xi})$  (respectively  $\hat{E}_2(x_1, \boldsymbol{\xi})$ ) we must keep only the eigenvectors corresponding to the negative (respectively the positive) real parts of the eigenvalues. Taking into account these considerations we replace the expressions of the local solutions into the interface conditions (5) to obtain the interface iterations on the  $\alpha$  coefficients:

$$\begin{cases} (\alpha_j^{1,p+1})_{j, \Re(\lambda_j) < 0}(\boldsymbol{\xi}) = \mathcal{T}_1 \left[ (\alpha_j^{2,p})_{j, \Re(\lambda_j) > 0}(\boldsymbol{\xi}) \right] \\ (\alpha_j^{2,p+1})_{j, \Re(\lambda_j) > 0}(\boldsymbol{\xi}) = \mathcal{T}_2 \left[ (\alpha_j^{1,p})_{j, \Re(\lambda_j) < 0}(\boldsymbol{\xi}) \right] \end{cases}$$

Then, the convergence rate of the  $\boldsymbol{\xi}$ -th component of the error vector of the Schwarz algorithm can be computed as the spectral radius of one of the iteration matrices  $\mathcal{T}_1 \mathcal{T}_2(\boldsymbol{\xi})$  or  $\mathcal{T}_2 \mathcal{T}_1(\boldsymbol{\xi})$ :

$$\rho_2^2 \equiv \rho_{\text{Schwarz2}}^2 = \rho(\mathcal{T}_1 \mathcal{T}_2) = \rho(\mathcal{T}_2 \mathcal{T}_1)$$

### 2.3 The 2D Euler equations

After having defined in a general frame the well-posedness of the boundary value problem associated to a general equation and the convergence of the Schwarz algorithm applied to this class of problems, we will concentrate ourselves on the conservative Euler equations in two-dimensions:

$$(7) \quad \frac{\partial W}{\partial t} + \nabla \cdot \mathbf{F}(W) = 0 \quad , \quad W = (\rho, \rho \mathbf{V}, E)^T \quad , \quad \nabla = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)^T .$$

In the above expressions,  $\rho$  is the density,  $\mathbf{V} = (u, v)^T$  is the velocity vector,  $E$  is the total energy per unit of volume and  $p$  is the pressure. In equation (7),  $W = W(\mathbf{x}, t)$  is the vector of conservative variables,  $\mathbf{x}$  and  $t$  respectively denote the space and time variables and  $\mathbf{F}(W) = (F_1(W), F_2(W))^T$  is the conservative flux vector whose components are given by

$$F_1(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{pmatrix}, \quad F_2(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{pmatrix}.$$

The pressure is deduced from the other variables using the state equation for a perfect gas  $p = (\gamma_s - 1)(E - \frac{1}{2} \rho \|\mathbf{V}\|^2)$  where  $\gamma_s$  is the ratio of the specific heats ( $\gamma_s = 1.4$  for the air).

### 2.4 A new type of interface conditions

We will apply now the method described previously to the computation of the convergence rate of the Schwarz algorithm applied to the two-dimensional subsonic Euler equations. In the supersonic case there is only one decomposition satisfying (3), that is:  $\mathcal{A}^{pos} = A_{\mathbf{n}}$  and  $\mathcal{A}^{neg} = 0$  and the convergence follows in 2 steps. Therefore the only case of interest is the subsonic one. The starting point of our analysis is given by the linearized form of the Euler equations (7) which are of the form (2) where we replace  $\delta W$

by  $W$  and to whom we applied a change of variable  $\tilde{W} = T^{-1}W$  based on the eigenvector factorization of  $A_1 = T\tilde{A}_1T^{-1}$ . In the following we will abandon the  $\tilde{\cdot}$  symbol):

$$\frac{W}{c\Delta t} + A_1\partial_x W + A_2\partial_y W = 0$$

characterized by the jacobian matrices  $A_1$  and  $A_2$  depending on  $M_n = \frac{u}{c}$ ,  $M_t = \frac{v}{c}$  which denote respectively the normal and the tangential Mach number. Before estimating the convergence rate we will derive the general transmission conditions at the interface by splitting the matrix  $A_1$  into a positive and negative part.

We have the following general result concerning this decomposition:

**Lemma 1** *Let  $\lambda_1 = M_n - 1$ ,  $\lambda_2 = M_n + 1$ ,  $\lambda_3 = \lambda_4 = M_n$ . Suppose we deal with a subsonic flow:  $0 < u < c$  so that  $\lambda_1 < 0$ ,  $\lambda_{2,3,4} > 0$ . Any decomposition of  $A_1 = A_{\mathbf{n}}$ ,  $\mathbf{n} = (1, 0)$  which satisfies (3) has to be of the form:*

$$\begin{aligned} \mathcal{A}^{neg} &= \frac{1}{a_1} \mathbf{u} \cdot \mathbf{u}^t, \mathbf{u} = (a_1, a_2, a_3, a_4)^t \\ \mathcal{A}^{pos} &= A_{\mathbf{n}} - \mathcal{A}^{neg}. \end{aligned}$$

where  $(a_1, a_2, a_3, a_4) \in \mathbb{R}^4$  satisfies  $a_1 \leq \lambda_1 < 0$  and  $\frac{a_1}{\lambda_1} + \frac{a_2^2}{a_1\lambda_2} + \frac{a_3^2}{a_1\lambda_3} + \frac{a_4^2}{a_1\lambda_4} = 1$ .

For the details of the proof see [DN04b].

We will proceed now to the estimation of the convergence rate using some results from [DLN04]. Following the technique described in section 2.2 we estimate the convergence rate in the non-overlapping case and we use the non-dimensioned wave-number  $\bar{\xi} = c\Delta t\xi$  and if we drop the bar symbol, we get for the general interface conditions the following:

$$(8) \quad \begin{cases} \rho_{2,novr}^2(\xi) &= \left| 1 - \frac{4M_n(1-M_n)(1+M_n)R(\xi)a_1^2(a+M_nR(\xi))}{D_1D_2} \right| \\ D_1 &= R(\xi)[a_1(1+M_n) - a_2(1-M_n)] + a[a_1(1+M_n) + a_2(1-M_n)] - i\sqrt{2}a_3\xi(1-M_n^2) \\ D_2 &= M_na_1[R(\xi)[a_1(1+M_n) - a_2(1-M_n)] + a[a_1(1+M_n) + a_2(1-M_n)]] \\ &\quad + a_3(1-M_n^2)[a_3(R+a) - iM_na_1\xi\sqrt{2}] \end{cases}$$

**Remark 2** *The expression (8) gives the convergence rate in the classical case for  $a_1 = -(1-M_n) = \lambda_1(0)$  and  $a_2 = a_3 = a_4 = 0$ , which corresponds to the classical transmission conditions. Moreover, theorem 1 proves that this quantity is always strictly inferior to 1 as the algorithm is convergent.*

In order to simplify our optimization problem we will take  $a_3 = 0$ , we can thus reduce the number of parameters to 2,  $a_1$  and  $a_2$ , as we can see that  $a_4$  can be expressed as a function of  $a_1$ ,  $a_2$  and  $a_3$ . We can also see that the convergence rate is a real quantity when the flow is normal to the interface  $M_t = 0$ . In the same time for the optimization purpose only we introduce the parameters:  $b_1 = -a_1/(1-M_n)$  and  $b_2 = a_2/(1+M_n)$  which provide a simpler form of the convergence rate:

$$(9) \quad \rho_{2,novr}^2(\xi) = \left| 1 - \frac{4b_1(a+M_nR(\xi))R(\xi)}{(R(\xi)(b_1+b_2) + a(b_1-b_2))^2(Mn+1)} \right|$$

Before proceeding to the analysis of the general case we recall some results found in the classical case obtained in [DLN04]. The asymptotic convergence rate in the non-overlapping case:

$$(10) \quad \lim_{k \rightarrow +\infty} \rho_{2,novr}(k) = \sqrt{\left(\frac{1-3M_n}{1+M_n}\right)^2 + \frac{8M_nM_t^2}{(1+M_n)^3}} < 1$$

is always strictly inferior to 1. Moreover, in the particular case  $M_n^* = 1/3$  and  $M_t = 0$ , this limit becomes null. The inequality (10) has a numerical meaning. For a given discretization, let  $\xi_{\max}$  denote the largest frequency supported by the numerical grid. This largest frequency is of the order  $\pi/h$  with  $h$  a typical mesh size. The convergence rate in a numerical computation made on this grid can be estimated by  $\rho_2^h = \max_{|\xi| < \xi_{\max}} \rho_2(\xi)$ . From (10), we have that  $\rho_2^h \leq \max_{|\xi| < \xi_{\max}} \rho_2(\xi) < 1$ . This means that for finer and finer grids, the number of iterations may increase slightly but should not go to infinity. Thus the optimization problem with respect to the parameters  $b_1$  and  $b_2$ , makes sense:

$$(11) \quad \min_{(b_1, b_2) \in \mathcal{I}_1 \times \mathcal{I}_2(b_1)} \max_{\xi \geq 0} \rho(\xi)$$

The solving of this problem is quite a tedious task even in the non-overlapping case, where we can obtain analytical expression of the parameters only for some values of the Mach number. In the same time, we have to analyze the convergence of the overlapping algorithm. Indeed, standard discretizations of the interface conditions correspond to overlapping decompositions with an overlap of size  $\delta = h$ ,  $h$  being the mesh size, as seen in [CFS98] and [DLN04]. Analytic optimization with respect to  $b_1$  and  $b_2$  seems out of reach. We will have to use numerical procedures of optimization.

In order to get closer to the numerical simulations we will estimate the convergence rate for the discretized equations with general transmission conditions, both in the non-overlapping and the overlapping case and then optimize numerically this quantity in order to get the best parameters for the convergence. Following a similar procedure as that described in [DLN04] and [DN04b] we will first discretize the problem by a finite volume scheme then we formulate a Schwarz algorithm whose convergence rate is estimated at the discrete level. Therefore, we will get the theoretical optimized parameters at the discrete level by means of a numerical algorithm, by calculating the following

$$(12) \quad \begin{aligned} \rho(b_1, b_2) &= \max_{k \in \mathcal{D}_h} \rho_2^2(k, \Delta x, M_n, M_t, b_1, b_2) \\ \min_{(b_1, b_2) \in \mathcal{I}_h} \rho(b_1, b_2) \end{aligned}$$

where  $\mathcal{D}_h$  is a uniform partition of the interval  $[0, \pi/\Delta x]$  and  $\mathcal{I}_h \subset \mathcal{I}$  a discretization by means of a uniform grid of a subset of the domain of the admissible values of the parameters. This kind of calculations are done once for all for a given pair  $(M_n, M_t)$  before the beginning of the Schwarz iterations. An example of such a result is given in the figure 1 Mach number  $M_n = 0.2$ . The computed parameters from the relation (12) will be further referred to with a superscript *th*. The theoretical estimates are compared afterwards with the numerical ones obtained by running the Schwarz algorithm with different pairs of parameters which lie in an interval such that the algorithm is convergent. We are thus able to estimate the optimal values for  $b_1$  and  $b_2$  from these numerical computations. These values will be referred to by a superscript *num*.

## 2.5 Implementation and numerical results

We present here a set of results of numerical experiments that are concerned with the evaluation of the influence of the interface conditions on the convergence of the non-overlapping Schwarz algorithm of the form. The computational domain is given by the rectangle  $[0, 1] \times [0, 1]$ . The numerical investigation is limited to the resolution of the linear system resulting from the first implicit time step using a Courant number CFL=100. In all these calculations we considered a model problem: a flow normal to the interface (that is when  $M_t = 0$ ). In figures 1 and 2 we can see an example of a theoretical and numerical estimation of the reduction factor of the error. We illustrate here the level curves which represent the log of the precision after 20 iterations for different values of the parameters  $(b_1, b_2)$ , the minimum being attained in this case for  $b_1^{th} = 1.3$  and  $b_2^{th} = -0.5$ ,  $b_1^{num} = 1.4$  and  $b_2^{num} = -0.6$ . We can see that we have good theoretical estimates of these parameters we can therefore use them in the interface conditions of the Schwarz algorithm.

Table 1: Overlapping Schwarz algorithm  
Numerical vs. theoretical parameters

$M_n$	$b_1^{th}$	$b_2^{th}$	$b_1^{num}$	$b_2^{num}$
0.1	1.6	-0.8	1.6	-0.9
0.2	1.3	-0.5	1.4	-0.6
0.3	1.25	-0.3	1.25	-0.45
0.4	1.08	-0.15	1.08	-0.28
0.5	1.03	-0.08	1.02	-0.23
0.6	1.0	0.0	1.0	0.0
0.7	1.02	0.06	1.01	0.04
0.8	1.03	0.08	1.02	0.06
0.9	1.06	0.08	1.04	0.06

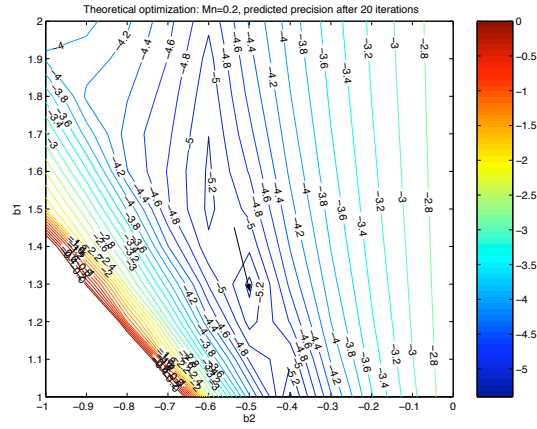


Figure 1: Isovalues of the predicted reduction factor of the error after 20 iterations via formula (12)

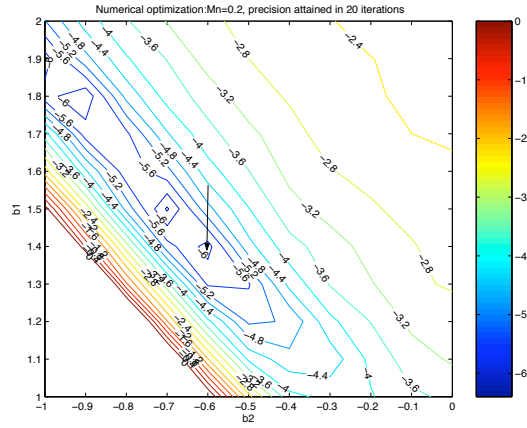


Figure 2: Isovalues of the reduction factor of the error after 20 iterations for the finite volume code

Table 2: Overlapping Schwarz algorithm  
Classical vs. optimized counts for different values of  $M_n$

$M_n$	$IT_0^{num}$	$IT_{op}^{num}$	$M_n$	$IT_0^{num}$	$IT_{op}^{num}$
0.1	48	19	0.5	22	18
0.2	41	20	0.7	20	16
0.3	32	20	0.8	22	15
0.4	26	19	0.9	18	12

Table 2 summarizes the number of Schwarz iterations required to reduce the initial linear residual by a factor  $10^{-6}$  for different values of the reference Mach number with the optimal parameters  $b_1^{num}$  and  $b_1^{num}$ . Here we denoted by  $IT_0^{num}$  and  $IT_{op}^{num}$  the observed (numerical) iteration number for classical and optimized interface conditions in order to achieve a convergence with a threshold  $\varepsilon = 10^{-6}$ . The same results are presented in figure 4. In the figure 3 we compare the theoretical estimated iteration number in the classical and optimized case. Comparing figures 3 and 4 we can see that the theoretical prediction are very close to the numerical tests.

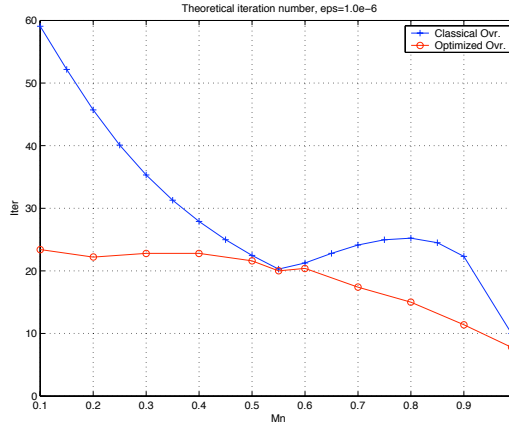


Figure 3: Theoretical iteration number: classical vs. optimized conditions

The conclusion of these numerical tests is, on one hand, that the theoretical prediction is very close to the numerical results: we can get by a numerical optimization (12) a very good estimate of optimal parameters ( $b_1, b_2$ ). On the other hand, the gain, in number of iterations, provided by the optimized interface conditions, is very promising for low Mach numbers, where the classical algorithm doesn't give optimal results. We can note that the optimized convergence rate is monotone with respect to the normal Mach number while the classical one isn't. For bigger Mach numbers, for instance, those who are close to 1, the classical algorithm already has a very good behaviour so the optimization is less useful. In the same time we studied here the zero order and therefore very simple transmission conditions. The use of higher order conditions (see [GMN02]) is a possible way that can be further studied to obtain even better convergence results.

### 3 A new preconditioning method

In this section we will show the equivalence between the linearized Euler system and a third order scalar equation. The motivation for this transformation is that a new algorithm is easier to design for a scalar



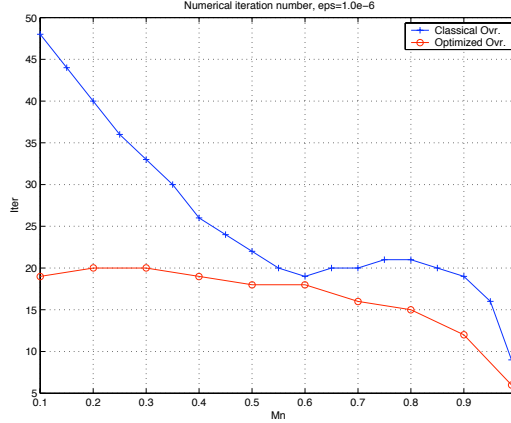


Figure 4: Numerical iteration number: classical vs. optimized conditions

equation than for a system of partial differential equations.

### 3.1 Equivalence of the Euler system to a scalar equation

The starting point of our analysis is given by the linearized form of the Euler equations (7) written in primitive variables  $(p, u, v, S)$ . In the following we suppose that the flow is isentropic, which allows us to drop the equation of the entropy (which is totally decoupled with respect to the others). We denote by  $W = (P, U, V)^T$  the vector of unknowns and by  $A$  and  $B$  the jacobian matrices of the fluxes  $F_i(w)$  to whom we already applied the variable change from conservative to primitive variables. In the following, we shall denote by  $\bar{c}$  the speed of the sound and we consider the linearized form (we will mark by the bar symbol, the constant state around which we linearize) of the Euler equations:

$$(13) \quad \mathcal{P}W \equiv \frac{W}{\Delta t} + A\partial_x W + B\partial_y W = f$$

characterized by the following jacobian matrices:

$$(14) \quad A = \begin{pmatrix} \bar{u} & \bar{\rho}\bar{c}^2 & 0 \\ 1/\bar{\rho} & \bar{u} & 0 \\ 0 & 0 & \bar{u} \end{pmatrix} \quad B = \begin{pmatrix} \bar{v} & 0 & \bar{\rho}\bar{c}^2 \\ 0 & \bar{v} & 0 \\ 1/\bar{\rho} & 0 & \bar{v} \end{pmatrix}$$

We can re-write the system (13) by denoting  $\beta = \frac{1}{\Delta t} > 0$  under the form

$$(15) \quad \mathcal{P}W \equiv (\beta I + A\partial_x + B\partial_y)W = f$$

We will study this system with the help of the Smith factorization.

#### 3.1.1 Smith factorization

We first recall the definition of the Smith factorization of a matrix with polynomial entries and apply it to systems of PDEs, see [Gan66] and references therein.

**Theorem 2** *Let  $n$  be an integer and  $C$  an invertible  $n \times n$  matrix with polynomial entries in the variable  $\lambda : C = (c_{ij}(\lambda))_{1 \leq i, j \leq n}$ .*

*Then, there exist three matrices with polynomial entries  $E$ ,  $D$  and  $F$  with the following properties:*

- $\det(E)=\det(F)=1$
- $D$  is a diagonal matrix.
- $C = EDF$ .

Moreover,  $D$  is uniquely defined up to a reordering and multiplication of each entry by a constant by a formula defined as follows. Let  $1 \leq k \leq n$ ,

- $S_k$  is the set of all the submatrices of order  $k \times k$  extracted from  $C$ .
- $Det_k = \{Det(B_k) \mid B_k \in S_k\}$
- $LD_k$  is the largest common divisor of the set of polynomials  $Det_k$ .

Then,

$$(16) \quad D_{kk}(\lambda) = \frac{LD_k(\lambda)}{LD_{k-1}(\lambda)}, \quad 1 \leq k \leq n$$

(by convention,  $LD_0 = 1$ ).

**Application to the Euler system** We first take formally the Fourier transform of the system (15) with respect to  $y$  (the dual variable is  $\xi$ ). We keep the partial derivatives in  $x$  since in the sequel we shall consider a domain decomposition with an interface whose normal is in the  $x$  direction. We note

$$(17) \quad \hat{\mathcal{P}} = \begin{pmatrix} \beta + \bar{u}\partial_x + i\xi\bar{v} & \bar{\rho}\bar{c}^2\partial_x & i\bar{\rho}\bar{c}^2\xi \\ \frac{1}{\bar{\rho}}\partial_x & \beta + \bar{u}\partial_x + i\xi\bar{v} & 0 \\ \frac{i\xi}{\bar{\rho}} & 0 & \beta + \bar{u}\partial_x + i\bar{v}\xi \end{pmatrix}$$

We can perform a Smith factorization of  $\hat{\mathcal{P}}$  by considering it as a matrix with polynomials in  $\partial_x$  entries. We have

$$(18) \quad \hat{\mathcal{P}} = EDF$$

where

$$(19) \quad D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \hat{\mathcal{L}}\hat{\mathcal{G}} \end{pmatrix}$$

and

$$E = \frac{1}{(\bar{u}(\bar{c}^2 - \bar{u}^2))^{1/3}} \begin{pmatrix} i\bar{\rho}\bar{c}^2\xi & 0 & 0 \\ 0 & \bar{u} & 0 \\ \beta + \bar{u}\partial_x + i\bar{v}\xi & E_2 & \frac{\bar{c}^2 - \bar{u}^2}{i\xi\bar{\rho}\bar{c}^2} \end{pmatrix}$$

and

$$(20) \quad F = - \begin{pmatrix} \frac{\beta + \bar{u}\partial_x + i\xi\bar{v}}{i\xi\bar{\rho}\bar{c}^2} & \frac{\partial_x}{i\xi} & 1 \\ \frac{\partial_x}{\bar{\rho}\bar{u}} & \frac{\beta + \bar{u}\partial_x + i\xi\bar{v}}{\bar{u}} & 0 \\ \frac{\bar{u}}{\beta + i\xi\bar{v}} & \frac{\bar{\rho}\bar{u}^2}{\beta + i\xi\bar{v}} & 0 \end{pmatrix}$$

where

$$E_2 = \bar{u} \frac{(-\bar{u}\bar{c}^2 + \bar{u}^3)\partial_{xx} + (2\bar{u}^2 - \bar{c}^2)(\beta + i\xi\bar{v})\partial_x + \bar{u}((\beta + i\xi\bar{v})^2 + \xi^2\bar{c}^2)}{\bar{c}^2(i\beta + i\xi\bar{v})},$$

$$(21) \quad \hat{\mathcal{G}} = \beta + \bar{u}\partial_x + i\xi\bar{v}$$

and

$$(22) \quad \hat{\mathcal{L}} = \beta^2 + 2i\xi\bar{u}\bar{v}\partial_x + 2\beta(\bar{u}\partial_x + i\xi\bar{v}) + (\bar{c}^2 - \bar{v}^2)\xi^2 - (\bar{c}^2 - \bar{u}^2)\partial_{xx}$$

Equation (19) suggests that the derivation of a domain decomposition method (DDM) for the third order operator  $\mathcal{LG}$  is a key ingredient for a DDM for the compressible Euler equations.

### 3.2 A new algorithm applied to a scalar third order problem

In this section we will describe a new algorithm applied to the third order operator found in section 3.1.1. We want to solve

$$(23) \quad \mathcal{LG}(Q) = g$$

where  $Q$  is scalar unknown function and  $g$  is a given right hand side. The algorithm will be based on the Robin-Robin algorithm [ATNV00] for the convection-diffusion problem. Then we will prove its convergence in 2 iterations. We first note that the elliptic operator  $\mathcal{L}$  can also be written as:

$$(24) \quad \mathcal{L} = -\text{div}(A\nabla) + \mathbf{a}\nabla + \beta^2, \quad A = \begin{pmatrix} \bar{c}^2 - \bar{u}^2 & -\bar{u}\bar{v} \\ -\bar{u}\bar{v} & \bar{c}^2 - \bar{v}^2 \end{pmatrix} \quad \text{where } \mathbf{a} = 2\beta(\bar{u}, \bar{v})$$

Without loss of generality we assume in the sequel that the flow is subsonic and that  $\bar{u} > 0$  and thus we have  $0 < \bar{u} < \bar{c}$ .

#### 3.2.1 The algorithm for a two-domain decomposition

We consider now a decomposition of the plane  $\mathbb{R}^2$  into two non-overlapping sub-domains  $\Omega_1 = (-\infty, 0) \times \mathbb{R}$  and  $\Omega_2 = (0, \infty) \times \mathbb{R}$ . The interface is  $\Gamma = \{x = 0\}$ . The outward normal to domain  $\Omega_i$  is denoted  $\mathbf{n}_i$ ,  $i = 1, 2$ . Let  $Q^{i,k}$ ,  $i = 1, 2$  represent the approximation to the solution in subdomain  $i$  at the iteration  $k$  of the algorithm. We define the following algorithm:

**ALGORITHM 1** We choose the initial values  $Q^{1,0}$  and  $Q^{2,0}$  such that  $\mathcal{G}Q^{1,0} = \mathcal{G}Q^{2,0}$  and we compute  $(Q^{i,k+1})_{i=1,2}$  from  $(Q^{i,k})_{i=1,2}$  by the following iterative procedure:

**Correction step** We compute the corrections  $\tilde{Q}^{1,k}$  and  $\tilde{Q}^{2,k}$  as solution of the homogeneous local problems:

$$(25) \quad \begin{cases} \mathcal{LG}\tilde{Q}^{1,k} = 0 & \text{in } \Omega_1, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}\tilde{Q}^{1,k} \cdot \mathbf{n}_1 = \gamma^k, & \text{on } \Gamma. \end{cases} \quad \begin{cases} \mathcal{LG}\tilde{Q}^{2,k} = 0 & \text{in } \Omega_2, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}\tilde{Q}^{2,k} \cdot \mathbf{n}_2 = \gamma^k, & \text{on } \Gamma, \\ \tilde{Q}^{2,k} = 0, & \text{on } \Gamma. \end{cases}$$

where  $\gamma^k = -\frac{1}{2} [A\nabla\mathcal{G}Q^{1,k} \cdot \mathbf{n}_1 + A\nabla\mathcal{G}Q^{2,k} \cdot \mathbf{n}_2]$ .

**Update step.** We update  $Q^{1,k+1}$  and  $Q^{2,k+1}$  by solving the local problems:

$$(26) \quad \begin{cases} \mathcal{LG}Q^{1,k+1} = g, & \text{in } \Omega_1, \\ \mathcal{G}Q^{1,k+1} = \mathcal{G}Q^{1,k} + \delta^k, & \text{on } \Gamma. \end{cases} \quad \begin{cases} \mathcal{LG}\tilde{Q}^{2,k+1} = g, & \text{in } \Omega_2, \\ \mathcal{G}Q^{2,k+1} = \mathcal{G}Q^{2,k} + \delta^k, & \text{on } \Gamma, \\ Q^{2,k+1} = Q^{1,k} + \tilde{Q}^{1,k}, & \text{on } \Gamma. \end{cases}$$

where  $\delta^k = \frac{1}{2} [\mathcal{G}\tilde{Q}^{1,k} + \mathcal{G}\tilde{Q}^{2,k}]$ .

**Proposition 1** Algorithm 1 converges in 2 iterations.

For the details of the proof see [DN04a].

### 3.3 A new algorithm applied to the Euler system

After having found an optimal algorithm which converges in two steps for the third order model problem, we focus on the Euler system by translating this algorithm into an algorithm for the Euler system. It suffices to replace the operator  $\mathcal{L}\mathcal{G}$  by the Euler system and  $Q$  by the last component  $F(W)_3$  of  $F(W)$  in the boundary conditions. This algorithm is quite complex since it involves second order derivatives of the unknowns in the boundary conditions on  $\mathcal{G}F(W)_3$ . It is possible to simplify it. By using the Euler equations in the subdomain, we have lowered the degree of the derivatives in the boundary conditions. After lengthy computations that we omit here, we find a simpler algorithm. We write it for a decomposition in two subdomains with an outflow velocity at the interface of domain  $\Omega_1$  but with an interface not necessarily rectilinear. In this way, it is possible to figure out how to use for a general domain decomposition. In the sequel,  $\mathbf{n} = (n_x, n_y)$  denotes the outward normal to domain  $\Omega_1$ ,  $\partial_n = \nabla \cdot \mathbf{n}$  the normal derivative at the interface,  $\partial_\tau = \nabla \cdot \tau$  the tangential derivative,  $U_n = U n_x + V n_y$  and  $U_\tau = -U n_y + V n_x$  are respectively the normal and tangential velocity at the interface between the subdomains. Similarly, we denote  $\bar{u}_n$  (resp.  $\bar{u}_\tau$ ) the normal (resp. tangential) component of the velocity around which we have linearized the equations.

**ALGORITHM 2** We choose the initial values  $W^{i,0} = (P^{i,0}, U^{i,0}, V^{i,0})$ ,  $i = 1, 2$  such that  $P^{1,0} = P^{2,0}$  and we compute  $W^{i,k+1}$  from  $W^{i,k}$  by the iterative procedure with two steps:

**Correction step** We compute the corrections  $\tilde{W}^{1,k}$  and  $\tilde{W}^{2,k}$  as solution of the homogeneous local problems:

$$(27) \quad \begin{cases} \mathcal{P}\tilde{W}^{1,k} = 0, & \text{in } \Omega_1, \\ -(\beta + \bar{u}_\tau \partial_\tau) \tilde{U}_n^{1,k} + \bar{u}_n \partial_\tau \tilde{U}_\tau^{1,k} = \gamma^k, & \text{on } \Gamma. \end{cases} \quad \begin{cases} \mathcal{P}\tilde{W}^{2,k} = 0, & \text{in } \Omega_2, \\ (\beta + \bar{u}_\tau \partial_\tau) \tilde{U}_n^{2,k} - \bar{u}_n \partial_\tau \tilde{U}_\tau^{2,k} = \gamma^k, & \text{on } \Gamma \\ \tilde{P}^{2,k} + \bar{\rho} \bar{u}_n \tilde{U}_n^{2,k} = 0, & \text{on } \Gamma. \end{cases}$$

where  $\gamma^k = -\frac{1}{2} \left[ (\beta + \bar{u}_\tau \partial_\tau) (U_n^{2,k} - U_n^{1,k}) + \bar{u}_n \partial_\tau (\tilde{U}_\tau^{1,k} - \tilde{U}_\tau^{2,k}) \right]$ .

**Update step.** We compute the update of the solution  $W^{1,k+1}$  and  $W^{2,k+1}$  as solution of the local problems:

$$(28) \quad \begin{cases} \mathcal{P}W^{1,k+1} = f_1, & \text{in } \Omega_1, \\ P^{1,k+1} = P^{1,k} + \delta^k, & \text{on } \Gamma. \end{cases} \quad \begin{cases} \mathcal{P}W^{2,k+1} = f_2, & \text{in } \Omega_2, \\ P^{2,k+1} = P^{2,k} + \delta^k, & \text{on } \Gamma, \\ (P + \bar{\rho} \bar{u}_n U_n)^{2,k+1} = (P + \bar{\rho} \bar{u}_n U_n)^{1,k} + (\tilde{P} + \bar{\rho} \bar{u}_n \tilde{U}_n)^{1,k}, & \text{on } \Gamma. \end{cases}$$

where  $\delta^k = \frac{1}{2} \left[ \tilde{P}^{1,k} + \tilde{P}^{2,k} \right]$ .

**Proposition 2** For a domain  $\Omega = \mathbb{R}^2$  divided into two non overlapping half planes, algorithm 2 converges in two iterations.

For the details of the proof see [DN04a].

### 3.4 Numerical results

We compare the proposed method and the classical method defined in [DLN04] involving interface conditions that are derived naturally from a weak formulation of the underlying boundary value problem. We present here a set of results of numerical experiments on a model problem. We consider a decomposition into different number of subdomains and for a linearization around a constant or non-constant flow. The computational domain is given by the rectangle  $[0, 4] \times [0, 1]$  with a uniform discretization using  $80 \times 20$  points. The numerical investigation is limited to the solving of the linear system resulting from the first implicit time step using a Courant number CFL=100. For all tests, the stopping criterion was a reduction of the maximum norm of the error by a factor  $10^{-6}$ . In the following, for the new algorithm, each

$M_n$	Classical iter.	Classical prec. GMRES	New DDM iter.	New prec. GMRES
0.001	32	18	16	16
0.01	30	18	16	16
0.1	28	17	14	14
0.2	24	17	14	14
0.3	20	17	14	14
0.4	18	16	14	14
0.5	16	16	12	12
0.6	15	16	12	12
0.7	14	16	12	12
0.8	14	17	14	14

Table 3: Subdomain solves counts for different values of  $M_n$ ,  $M_t(y)$

$h$ ( $M_n = 0.001$ )	Classical	New DDM	$h$ ( $M_n = 0.1$ )	Classical	New DDM
1/10	65	18	1/10	56	12
1/20	67	18	1/20	57	14
1/40	70	18	1/40	59	16

Table 4: Subdomain solves counts for different mesh size

iteration counts for 2 as we need to solve twice as much local problems than with the classical algorithm. For an easier comparison of the algorithms, the figures shown in the tables are the number of subdomains solves.

In Table 3, we consider a linearization around a variable state where the tangential velocity is given by  $M_t(y) = 0.1(1 + \cos(\pi y))$  and we vary the normal Mach number. In figure 5, we linearize the equations around a variable state for a normal flow to the interface ( $M_t = 0.0$ ), where the initial normal velocity is gives by  $M_n(y) = 0.5(0.2 + 0.04 \tanh(y/0.2))$ . The sensitivity to the mesh size is shown in the Table 4.

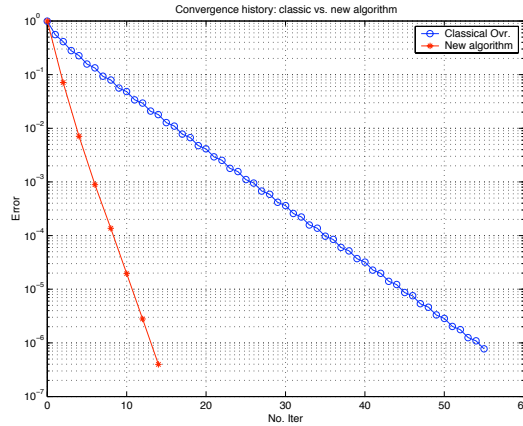


Figure 5: Convergence curves for the classical and the new algorithms

We can see that for the new algorithm the growth in the number of iterations is very weak as the mesh is refined, the same property being already known for the classical one.

In figure 6, we consider for a three subdomains decomposition a linearization around a variable state

$M_n$	Classical (iter.)	Classical prec GMRES	New DDM prec. GMRES
0.001	75	36	24
0.01	70	36	24
0.1	56	36	30
0.2	44	36	32
0.3	34	34	34

Table 5: Iteration count for different values of  $M_n$

for a normal flow to the interface. The normal velocity is given by  $M_n(y) = 0.5(0.2 + 0.04 \tanh(y/0.2))$  (the same as for the 2 subdomain case).

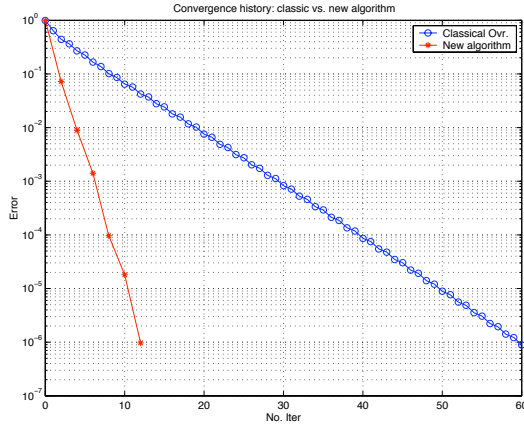


Figure 6: Convergence curves for the classical and the new algorithm

These tests show that the new algorithm is very stable with respect to various parameters such as the mesh size and the Mach number. We see that the convergence in two iterations of the continuous algorithm is lost at the discrete level although the subdomain solves are very reasonable. Moreover, a stabilization was necessary for the discretization of the interface condition (27) in order to keep the algorithm converging. The optimal discretization of this interface condition is not yet quite well known. The comparison with the classical algorithm is favorable for Mach numbers smaller than 0.5 and especially very low Mach numbers by a factor of almost 4.

The next set of tests concerns a decomposition into 4 subdomains using a  $2 \times 2$  decomposition of a  $40 \times 40 = 1600$  point mesh. The Table 5 summarizes the number of GMRES iterations required to reduce the initial linear residual by a factor  $10^{-6}$  for different values of the reference Mach number for the classical algorithm and the number of GMRES iteration necessary to achieve convergence when solving the interface system.

We now consider a linearization around a variable state for a normal flow to the interface, where the normal velocity is gives by the expression  $M_n(y) = 0.5(0.2 + 0.04 \tanh(y/0.2))$  (the same as for the 2 subdomain case) and we are solving the homogeneous equations verified by the error vector at the first time step. The convergence history is given in the figure 7.

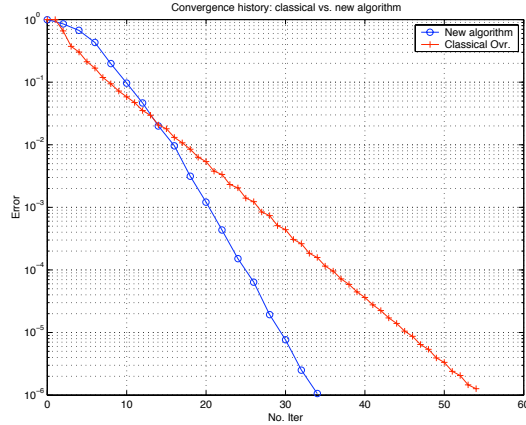


Figure 7: Comparison between the classical and the new algorithm

### 3.5 Conclusion

We designed a new domain decomposition for the Euler equations inspired by the idea of the Robin-Robin preconditioner applied to the advection-diffusion equation. We used the same principle after reducing the system to scalar equations via a Smith factorization. The resulting algorithm behaves very well for low Mach numbers, where usually the classical algorithm doesn't give very good results. We reduce the number of subdomain solves by almost a factor 4 for linearization around a constant and variable state as well. A general theoretical study and more comprehensive numerical tests have to be done in order to firmly assess the applicability of the proposed algorithm to large scale computations.

This work can also be seen as a first step for deriving new domain decomposition methods for the 2D or 3D compressible Navier-Stokes equations. Indeed, the derivation of the algorithm which is based on the Smith factorization is in fact general and can be applied to arbitrary systems of partial differential equations.

## References

- [ATNV00] Yves Achdou, Patric Le Tallec, Frédéric Nataf, and Marina Vidrascu. A domain decomposition preconditioner for an advection-diffusion problem. *Comput. Methods Appl. Mech. Engrg.*, 184:145–170, 2000.
- [BD97] J. D. Benamou and B. Després. A domain decomposition method for the Helmholtz equation and related optimal control. *J. Comp. Phys.*, 136:68–82, 1997.
- [CFS98] X.-C. Cai, C. Farhat, and M. Sarkis. A minimum overlap restricted additive Schwarz preconditioner and applications to 3D flow simulations. *Contemporary Mathematics*, 218:479–485, 1998.
- [Cle98] S. Clerc. Non-overlapping Schwarz method for systems of first order equations. *Cont. Math.*, 218:408–416, 1998.
- [CN98] Philippe Chevalier and Frédéric Nataf. Symmetrized method with optimized second-order conditions for the Helmholtz equation. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, pages 400–407. Amer. Math. Soc., Providence, RI, 1998.
- [DLN04] V. Dolean, S. Lanteri, and F. Nataf. Convergence analysis of a schwarz type domain decomposition method for the solution of the euler equations. *Appl. Num. Math.*, 49:153–186, 2004.

- [DN04a] V. Dolean and F. Nataf. A new domain decomposition method for the compressible euler equations. Technical Report 567, CMAP - Ecole Polytechnique, 2004.
- [DN04b] V. Dolean and F. Nataf. An optimized schwarz algorithm for the compressible euler equations. Technical Report 556, CMAP - Ecole Polytechnique, 2004.
- [EZ98] Bjorn Engquist and Hong-Kai Zhao. Absorbing boundary conditions for domain decomposition. *Appl. Numer. Math.*, 27(4):341–365, 1998.
- [Gan66] Felix R. Gantmacher. *Theorie des matrices*. Dunod, 1966.
- [GGTN04] L. Gerardo-Giorda, P. Le Tallec, and F. Nataf. A robin-robin preconditioner for advection-diffusion equations with discontinuous coefficients. *Comput. Methods Appl. Mech. Engrg.*, 193:745–764, 2004.
- [GHN01] Martin J. Gander, Laurence Halpern, and Frédéric Nataf. Optimal Schwarz waveform relaxation for the one dimensional wave equation. Technical Report 469, CMAP, Ecole Polytechnique, September 2001.
- [GKM<sup>+</sup>91] R. Glowinski, Y.A. Kuznetsov, G. Meurant, J. Periaux, and O.B. Widlund, editors. *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, 1991. SIAM.
- [GMN02] M.-J. Gander, F. Magoulès, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 24-1:38–60, 2002.
- [JNR01] Caroline Japhet, Frédéric Nataf, and Francois Rogier. The optimized order 2 method. application to convection-diffusion problems. *Future Generation Computer Systems FUTURE*, 18, 2001.
- [Lio90] Pierre-Louis Lions. On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In Tony F. Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989*, Philadelphia, PA, 1990. SIAM.
- [QS96] A. Quarteroni and L. Stolicis. Homogeneous and heterogeneous domain decomposition methods for compressible flow at high reynolds numbers. Technical Report 33, CRS4, 1996.
- [RT91] Y.H. De Roeck and P. Le Tallec. Analysis and Test of a Local Domain Decomposition Preconditioner. In *R. Glowinski et al. [GKM<sup>+</sup>91]*, 1991.
- [TW04] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*. Springer Series in Computational Mathematics. Springer Verlag, 2004.